

INTEGRATED SYMBOLIC CONTROL DESIGN FOR NONLINEAR SYSTEMS WITH INFINITE STATES SPECIFICATIONS

GIORDANO POLA¹, ALESSANDRO BORRI¹ AND MARIA D. DI BENEDETTO¹

ABSTRACT. Discrete abstractions of continuous and hybrid systems have recently been the topic of great interest from both the control systems and the computer science communities, because they provide a sound mathematical framework for analysing and controlling embedded systems. In this paper we give a further contribution to this research line, by addressing the problem of symbolic control design of nonlinear systems with infinite states specifications, modelled by differential equations. We first derive the symbolic controller solving the control design problem, given in terms of discrete abstractions of the plant and the specification systems. We then present an algorithm which integrates the construction of the discrete abstractions with the design of the symbolic controller. Space and time complexity analysis of the proposed algorithm is performed and a comparison with traditional approaches currently available in the literature for symbolic control design, is discussed. Some examples are included, which show the interest and applicability of our results.

1. INTRODUCTION

Discrete abstractions of continuous and hybrid systems have been the topic of intensive study in the last twenty years from both the control systems and the computer science communities [EFP06]. While physical world processes are often described by differential equations, digital controllers and software and hardware at the implementation layer, are usually modelled through discrete/symbolic processes. This mathematical models heterogeneity has posed during the years interesting and challenging theoretical problems that are needed to be addressed, in order to ensure the formal correctness of control algorithms. One approach to deal with this heterogeneity is to construct symbolic models that are equivalent to the continuous process, so that the mathematical model of the process, of the controller, and of the software and hardware at the implementation layer, are of the same nature. Several classes of dynamical and control systems admitting symbolic models, were identified during the years. We recall timed automata [AD94], rectangular hybrid automata [HKPV98], and o-minimal hybrid systems [LPS00] in the class of hybrid automata. Control systems were considered further. Early results in this regard are reported in the work of [CW98], [MRO02], [FJL02] and [BMP02]. Recent results include the work of [TP06], which showed existence of symbolic models for controllable discrete-time linear systems, and the work of [HCS06, BH06] for piecewise-affine and multi-affine systems. Many of the aforementioned work are based on the notion of bisimulation equivalence, introduced by Milner and Park [Mil89, Par81] in the context of concurrent processes, as a formal equivalence notion to relate continuous and hybrid processes to purely discrete/symbolic models. A new insight in the construction of symbolic models has been recently placed through the notion of approximate bisimulation introduced by Girard and Pappas in [GP07]. Based on the above notion, some classes of incrementally stable [Ang02] control systems were recently shown to admit symbolic models: discrete-time linear control systems [Gir07], nonlinear control systems with and without disturbances [PGT08, PT09], nonlinear time-delay systems [PPDT10] and switched nonlinear systems [GPT10]. Recent results in the work of [ZPT10] have also shown the existence of symbolic models for unstable nonlinear control systems, satisfying the so-called incremental forward completeness property. The use of symbolic models in the control design of continuous and hybrid systems has been investigated in the work of [TP06, YB09, Tab08], among many others. The work in [TP06] considers discrete-time linear control systems, the work in [YB09] considers piecewise-affine systems while the work in [Tab08] considers stabilizable nonlinear control systems. In this paper we give a further contribution to this research line and in

This work has been partially supported by the Center of Excellence for Research DEWS, University of L'Aquila, Italy.

particular, in the direction of [Tab08]. We consider symbolic control design of nonlinear control systems where specifications are characterized by an infinite number of states and modelled through differential equations: Given a plant nonlinear control system and a specification nonlinear (autonomous) system, we investigate conditions for the existence of a symbolic controller that implements the behaviour of the specification, with a precision that can be rendered as small as desired. In other words, we look for a symbolic controller so that the interconnection between the plant and the controller satisfies or conforms [CGP99] the specification with an arbitrarily small precision. The symbolic controller is furthermore requested to be non-blocking in order to prevent the occurrence of deadlocks in the interaction between the plant and the symbolic controller. This control design problem can be seen as an approximated version of similarity games, as discussed in [Tab09]. Similar problems have been studied in the literature (in a non-approximating settings) in the context of supervisory control [CL99], symbolic control design for piecewise-affine systems enforcing temporal logic specifications [YB09], among many others.

The control design problem that we consider in this paper has been solved by following the so-called correct-by-design approach, see e.g. [TP06, Tab08, YB09]. We first construct the symbolic models of the plant and the specification by making use of (some variations of) the results established in [PGT08]. We then solve the control design problem at the symbolic layer, to finally come back at the continuous layer, by providing appropriate approximating bounds in the quantization errors which guarantee the solution to the control design problem under study. The solution of the control design problem at the symbolic layer is shown to be the maximal non-blocking part of the (exact) parallel composition [CL99] of the symbolic models associated with the plant and the specification. By following the correct-by-design approach, the design of the symbolic controller solving the problem at hand, requires a first computation of the plant and the specification symbolic models, then a construction of the (exact) parallel composition of the symbolic systems obtained and finally a computation of the maximal non-blocking part of the composed system. While being formally correct from the theoretical point of view, this approach is in general rather demanding from the computational point of view, because of the large size of the symbolic models needed to be constructed, in order to synthesize the symbolic controller solving the design problem. This drawback is common with other approaches currently available in the literature on symbolic control design of continuous and hybrid systems, see e.g. [TP06, YB09, Tab08] and motivated some researchers to propose solutions to cope with complexity. For example, the work in [TiI09] proposes nonuniform state quantizations in the construction of the symbolic models of the to-be-controlled plant system. In this paper we propose an alternative solution to the one studied in [TiI09]. Inspired by on-the-fly verification and control of timed or untimed transition systems (see e.g. [CVWY92, TA99]), we approach the design of symbolic controllers by advocating an “integration” philosophy: instead of computing separately the symbolic models of the plant and of the specification to then design the controller at the symbolic layer, *we integrate each step of the procedure in only one algorithm*. Space and time complexity analysis of the proposed algorithm is performed and a comparison with traditional approaches currently available in the literature, is discussed. Some examples are included which show the interest and applicability of our results. For the sake of completeness, a detailed list of the employed notation is included in the Appendix (Section 8).

2. PRELIMINARY DEFINITIONS

2.1. Control Systems. The class of control systems that we consider in this paper is formalized in the following definition.

Definition 2.1. A *control system* is a quintuple:

$$(2.1) \quad \Sigma = (X, X_0, U, \mathcal{U}, f),$$

where:

- $X \subseteq \mathbb{R}^n$ is the state space;
- $X_0 \subseteq X$ is the set of initial states;
- $U \subseteq \mathbb{R}^m$ is the input space;

- \mathcal{U} is a subset of the set of all locally essentially bounded functions of time from intervals of the form $]a, b[\subseteq \mathbb{R}$ to U with $a < 0$, $b > 0$;
- $f : \mathbb{R}^n \times U \rightarrow \mathbb{R}^n$ is a continuous map satisfying the following Lipschitz assumption: for every compact set $K \subset \mathbb{R}^n$, there exists a constant $\kappa \in \mathbb{R}^+$ such that

$$\|f(x, u) - f(y, u)\| \leq \kappa \|x - y\|,$$

for all $x, y \in K$ and all $u \in U$.

A curve $\xi :]a, b[\rightarrow \mathbb{R}^n$ is said to be a *trajectory* of Σ if there exists $u \in \mathcal{U}$ satisfying:

$$(2.2) \quad \dot{\xi}(t) = f(\xi(t), u(t)),$$

for almost all $t \in]a, b[$. Although we have defined trajectories over open domains, we shall refer to trajectories $\xi : [0, \tau] \rightarrow \mathbb{R}^n$ defined on closed domains $[0, \tau]$, $\tau \in \mathbb{R}^+$ with the understanding of the existence of a trajectory $\xi' :]a, b[\rightarrow \mathbb{R}^n$ such that $\xi = \xi'|_{[0, \tau]}$. We also write $\xi_{xu}(\tau)$ to denote the point reached at time τ under the input u from initial condition x ; this point is uniquely determined, since the assumptions on f ensure existence and uniqueness of trajectories [Son98]. A control system Σ is said to be forward complete if every trajectory is defined on an interval of the form $]a, \infty[$. Sufficient and necessary conditions for a system to be forward complete can be found in [AS99]. The above formulation of control systems can be also used to model autonomous nonlinear systems, i.e. systems with no control inputs. With a slight abuse of notation we denote an autonomous system Σ by means of the tuple (X, X_0, f) .

2.2. Systems. We will use systems to describe both control systems as well as their symbolic models. For a detailed exposition of the notion of systems and of their properties we refer to [Tab09].

Definition 2.2. [Tab09] A system S is a sextuple:

$$S = (X, X_0, U, \longrightarrow, Y, H),$$

consisting of:

- a set of states X ;
- a set of initial states $X_0 \subseteq X$;
- a set of inputs U ;
- a transition relation $\longrightarrow \subseteq X \times U \times X$;
- an output set Y ;
- an output function $H : X \rightarrow Y$.

A transition $(x, u, x') \in \longrightarrow$ of system S is denoted by $x \xrightarrow{u} x'$. System S is said to be:

- *countable*, if X and U are countable sets;
- *symbolic*, if X and U are finite sets;
- *metric*, if the output set Y is equipped with a metric $d : Y \times Y \rightarrow \mathbb{R}_0^+$;
- *deterministic*, if for any $x \in X$ and $u \in U$ there exists at most one $x' \in X$ such that $(x, u, x') \in \longrightarrow$;
- *non-blocking*, if for any $x \in X$ there exists $(x, u, x') \in \longrightarrow$;
- *accessible*, if for any $x \in X$ there exists a finite number of transitions

$$x_0 \xrightarrow{u_1} x_1 \xrightarrow{u_2} \dots \xrightarrow{u_N} x$$

starting from an initial state x_0 in X_0 and ending up in x .

We now introduce some notions which will be employed in the further developments. We start by introducing the notion of sub-system which formalizes the idea of extracting from the original system a subset of states, inputs and transitions.

Definition 2.3. Given two systems $S_1 = (X_1, X_{0,1}, U_1, \xrightarrow{1}, Y_1, H_1)$ and $S_2 = (X_2, X_{0,2}, U_2, \xrightarrow{2}, Y_2, H_2)$, system S_1 is a sub-system of S_2 , denoted $S_1 \sqsubseteq S_2$, if $X_1 \subseteq X_2$, $X_{0,1} \subseteq X_{0,2}$, $U_1 \subseteq U_2$, $\xrightarrow{1} \subseteq \xrightarrow{2}$, $Y_1 \subseteq Y_2$ and $H_1(x) = H_2(x)$ for any $x \in X_1$.

The following notion formalizes the idea of extracting the maximal non-blocking sub-system from a system, where maximality is given with respect to the notion of sub-system, which naturally induces a preorder on the class of systems.

Definition 2.4. Given a system $S = (X, X_0, U, \longrightarrow, Y, H)$ the non-blocking part of S is a system $Nb(S)$ so that:

- (i) $Nb(S)$ is a non-blocking system;
- (ii) $Nb(S)$ is a sub-system of S ;
- (iii) $S' \sqsubseteq Nb(S)$, for any non-blocking $S' \sqsubseteq S$.

We finally introduce the notion of accessible part [CL99] which formalizes the idea of extracting the maximal accessible sub-system from a system.

Definition 2.5. Given a system $S = (X, X_0, U, \longrightarrow, Y, H)$ the accessible part of S is a system $Ac(S)$ so that:

- (i) $Ac(S)$ is an accessible system;
- (ii) $Ac(S)$ is a sub-system of S ;
- (iii) $S' \sqsubseteq Ac(S)$, for any accessible $S' \sqsubseteq S$.

In this paper we consider simulation and bisimulation relations [Mil89, Par81] that are useful when analyzing or designing controllers for deterministic systems [Tab09]. Bisimulation relations are standard mechanisms to relate the properties of systems. Intuitively, a bisimulation relation between a pair of systems S_1 and S_2 is a relation between the corresponding state sets explaining how a state trajectory s_1 of S_1 can be transformed into a state trajectory s_2 of S_2 and vice versa. While typical bisimulation relations require that s_1 and s_2 are observationally indistinguishable, that is $H_1(s_1) = H_2(s_2)$, we shall relax this by requiring $H_1(s_1)$ to simply be close to $H_2(s_2)$ where closeness is measured with respect to the metric on the output set. A simulation relation is a one-sided version of a bisimulation relation. The following notions have been introduced in [GP07] and in a slightly different formulation in [Tab08].

Definition 2.6. Let $S_1 = (X_1, X_{0,1}, U_1, \xrightarrow{1}, Y_1, H_1)$ and $S_2 = (X_2, X_{0,2}, U_2, \xrightarrow{2}, Y_2, H_2)$ be metric systems with the same output sets $Y_1 = Y_2$ and metric d , and consider a precision $\varepsilon \in \mathbb{R}_0^+$. A relation

$$\mathcal{R} \subseteq X_1 \times X_2,$$

is said to be an ε -approximate simulation relation from S_1 to S_2 , if the following conditions are satisfied:

- (i) for every $x_1 \in X_{0,1}$, there exists $x_2 \in X_{0,2}$ with $(x_1, x_2) \in \mathcal{R}$;
- (ii) for every $(x_1, x_2) \in \mathcal{R}$ we have $d(H_1(x_1), H_2(x_2)) \leq \varepsilon$;
- (iii) for every $(x_1, x_2) \in \mathcal{R}$ we have that:

$$x_1 \xrightarrow{u_1} x'_1 \text{ in } S_1 \text{ implies the existence of } x_2 \xrightarrow{u_2} x'_2 \text{ in } S_2 \text{ satisfying } (x'_1, x'_2) \in \mathcal{R}.$$

System S_1 is ε -approximately simulated by S_2 or S_2 ε -approximately simulates S_1 , denoted by $S_1 \preceq_\varepsilon S_2$, if there exists an ε -approximate simulation relation from S_1 to S_2 . When $\varepsilon = 0$, system S_1 is said to be 0-simulated by S_2 or S_2 is said to 0-simulate S_1 .

By symmetrizing the notion of approximate simulation we obtain the notion of approximate bisimulation, which is reported hereafter.

Definition 2.7. Let $S_1 = (X_1, X_{0,1}, U_1, \xrightarrow{1}, Y_1, H_1)$ and $S_2 = (X_2, X_{0,2}, U_2, \xrightarrow{2}, Y_2, H_2)$ be metric systems with the same output sets $Y_1 = Y_2$ and metric d , and consider a precision $\varepsilon \in \mathbb{R}_0^+$. A relation

$$\mathcal{R} \subseteq X_1 \times X_2,$$

is said to be an ε -approximate bisimulation relation between S_1 and S_2 , if the following conditions are satisfied:

- (i) \mathcal{R} is an ε -approximate simulation relation from S_1 to S_2 ;
- (ii) \mathcal{R}^{-1} is an ε -approximate simulation relation from S_2 to S_1 .

System S_1 is ε -approximately bisimilar to S_2 , denoted by $S_1 \cong_\varepsilon S_2$, if there exists an ε -approximate bisimulation relation \mathcal{R} between S_1 and S_2 . When $\varepsilon = 0$, system S_1 is said to be 0-bisimilar or exactly bisimilar to S_2 .

We now introduce the notion of approximate composition of systems which is employed in the further developments to formalize the interconnection between a nonlinear control system representing the plant, and a symbolic system representing the symbolic controller.

Definition 2.8. [Tab08] Given two metric systems $S_1 = (X_1, X_{0,1}, U_1, \xrightarrow{1}, Y_1, H_1)$ and $S_2 = (X_2, X_{0,2}, U_2, \xrightarrow{2}, Y_2, H_2)$, with the same output sets $Y_1 = Y_2$ and metric d and a precision $\varepsilon \in \mathbb{R}_0^+$, the ε -approximate composition of S_1 and S_2 is the system:

$$S_1 \parallel_\varepsilon S_2 := (X, X_0, U, \longrightarrow, Y, H),$$

where:

- $X = \{(x_1, x_2) \in X_1 \times X_2 : d(H_1(x_1), H_2(x_2)) \leq \varepsilon\}$;
- $X_0 = X \cap (X_{0,1} \times X_{0,2})$;
- $U = U_1 \times U_2$;
- $(x_1, x_2) \xrightarrow{(u_1, u_2)} (x'_1, x'_2)$ if $x_1 \xrightarrow{u_1} x'_1$ and $x_2 \xrightarrow{u_2} x'_2$;
- $Y = Y_1$;
- $H : X_1 \times X_2 \rightarrow Y$ is given by $H(x_1, x_2) := H_1(x_1)$, for any $(x_1, x_2) \in X$.

The above notion of composition is asymmetric. This is because it models the interaction of systems S_1 and S_2 which play different roles in the composition. As it will be clarified in the next section, we interpret system S_1 as the plant system, i.e. the to-be-controlled process, and system S_2 as the controller.

3. PROBLEM STATEMENT

In this paper we address the problem of symbolic control design for nonlinear systems with infinite states specifications modelled by differential equations. In order to formally define the control design problem under consideration, we first need to provide a formal notion of symbolic controllers. Given a control system $\Sigma = (X, X_0, U, \mathcal{U}, f)$ and a sampling time parameter $\tau \in \mathbb{R}^+$, we associate the following system to Σ :

$$(3.1) \quad S_\tau(\Sigma) := (X, X_0, \mathcal{U}_\tau, \xrightarrow{\tau}, Y, H),$$

where:

- $\mathcal{U}_\tau = \{u \in \mathcal{U} \mid \text{the domain of } u \text{ is } [0, \tau]\}$;
- $x \xrightarrow{\tau} x'$ if there exists a trajectory $\xi : [0, \tau] \rightarrow X$ of Σ satisfying $\xi_{xu}(\tau) = x'$;
- $Y = X$;
- $H = 1_X$.

System $S_\tau(\Sigma)$ is metric when we regard $Y = X$ as being equipped with the metric $d(p, q) = \|p - q\|$. The above system can be thought of as the time discretization of the control system Σ .

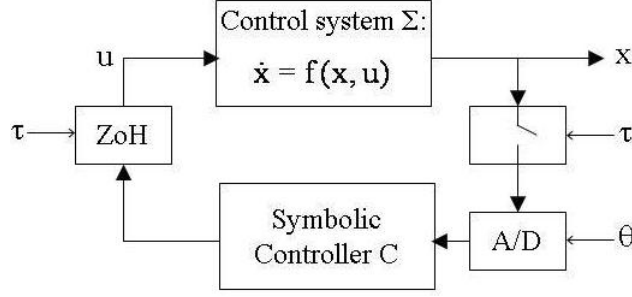


FIGURE 1. Approximate composition of the plant and specification systems.

Definition 3.1. Given the control system Σ , a sampling time $\tau \in \mathbb{R}^+$, a state quantization $\theta \in \mathbb{R}^+$ and an input quantization $\mu \in \mathbb{R}^+$, a *symbolic controller* for Σ is formalized by means of the system:

$$C := (X_c, X_{c,0}, U_c, \xrightarrow{c}, Y_c, H_c),$$

where¹:

- $X_c = [X]_{2\theta}$;
- $X_{c,0} \subseteq X_c$;
- $U_c = \{u \in \mathcal{U}_\tau \mid \text{the co-domain of } u \text{ is } [U]_{2\mu}\}$;
- $\xrightarrow{c} \subseteq X_c \times U_c \times X_c$;
- $Y_c = X_c$;
- $H_c = 1_{X_c}$.

We denote by $\mathcal{C}^{\tau, \theta, \mu}(\Sigma)$ the class of symbolic controllers with sampling time τ , state quantization θ and input quantization μ , associated with Σ . The θ -approximate composition between the time discretization $S_\tau(\Sigma)$ of a control system Σ and a symbolic controller $C \in \mathcal{C}^{\tau, \theta, \mu}(\Sigma)$ formalizes classical static state feedback control schemes with digital controllers, studied in the literature, see e.g. [FPW98], as illustrated in Figure 1: The state signal $\xi_{x_0 u}(t)$ at time $t \in \mathbb{R}^+$ is firstly sampled with sampling time $\tau \in \mathbb{R}^+$, then quantized through an Analog-to-Digital (A/D) converter with precision $\theta \in \mathbb{R}^+$ which associates to a state $\xi_{x_0 u}(\tau)$, the unique state $x \in X_c$ for which² $\xi_{x_0 u}(\tau) \in \mathcal{B}_{[\theta]}(x)$; the obtained digital/symbolic signal is then plugged as input to the digital/symbolic controller C which outputs a symbolic signal taking values in $[U]_{2\mu}$. Such symbolic signal is then plugged into a Zero order Holder (ZoH) with sampling time parameter τ which outputs in turn, a piecewise-constant signal u that is finally plugged as digital/symbolic control input to the control system Σ .

We are now ready to formally state the symbolic control design problem that we consider in this paper. Consider a plant nonlinear control system:

$$(3.2) \quad P = (X_p, X_{p,0}, U_p, \mathcal{U}_p, f_p),$$

and a specification nonlinear autonomous system:

$$Q = (X_q, X_{q,0}, g_q).$$

For the sake of homogeneity in the notation of the plant P and the specification Q we rephrase the above tuple by means of:

$$(3.3) \quad Q = (X_q, X_{q,0}, U_q, \mathcal{U}_q, f_q),$$

¹ The sets $[X]_{2\theta}$ and $[U]_{2\mu}$, are lattices embedded in the sets \mathbb{R}^n and U , with precisions θ and μ respectively, as formally defined in the Appendix.

²The set $\mathcal{B}_{[\theta]}(x)$ is defined in the Appendix.

where $U_q = \{u_q\}$ with $u_q = 0$, $\mathcal{U}_q = \{\mathbf{u}_q\}$ with $\mathbf{u}_q = \mathbf{0}$, the signal $\mathbf{0}$ being the identically null function, and $f_q(x, u) = g_q(x) + u$ for any $(x, u) \in X_q \times U_q$.

Problem 3.2. *Given a plant nonlinear control system P as in (3.2), a specification nonlinear autonomous system Q as in (3.3) and a desired precision $\varepsilon \in \mathbb{R}^+$, find quantization parameters $\tau, \theta, \mu \in \mathbb{R}^+$ and a symbolic controller $C \in \mathcal{C}^{\tau, \theta, \mu}(P)$ such that:*

- (i) $(S_\tau(P) \parallel_\theta C) \preceq_\varepsilon S_\tau(Q)$;
- (ii) $S_\tau(P) \parallel_\theta C$ is non-blocking.

The above control design problem asks for a symbolic controller C that implements the behaviour of the specification Q , up to a precision ε that can be chosen as small as desired. In other words, in Problem 3.2 we look for a symbolic controller C so that the approximate composition between the plant P and the controller C satisfies or conforms [CGP99] the specification Q with an arbitrarily small precision. The symbolic controller is furthermore requested to be non-blocking in order to prevent occurrence of deadlocks in the interaction between the plant and the symbolic controller. This control design problem can be seen as an approximated version of similarity games, as discussed in [Tab09]. Similar problems have been studied in the literature (in a non-approximating settings) in the context of supervisory control [CL99], symbolic control design for piecewise-affine systems enforcing temporal logic specifications [YB09], among many other work.

4. SYMBOLIC CONTROL DESIGN WITH INFINITE STATES SPECIFICATIONS

In this section we provide the solution to Problem 3.2. Inspired by the so-called correct-by design approach, see e.g. [TP06, Tab08, YB09], we first construct the symbolic systems associated with the plant P and the specification Q in Section 4.1, we then solve the control design problem at the symbolic layer in Section 4.2 to finally come back at the continuous layer in Section 4.3 by providing the bounds in the approximation scheme that we propose, which guarantee the solution to Problem 3.2.

4.1. From the Continuous Layer to the Symbolic Layer. In this section we present some results based on the work of [PGT08] for constructing symbolic systems associated with the plant P and the specification Q . We start by recalling from [Ang02], the notion of incremental input-to-state stability for nonlinear control systems.

Definition 4.1. A control system Σ is incrementally input-to-state stable (δ -ISS) if it is forward complete and there exist a \mathcal{KL} function β and a \mathcal{K}_∞ function γ such that for any $t \in \mathbb{R}_0^+$, any $x, x' \in \mathbb{R}^n$, and any $u, u' \in \mathcal{U}$ the following condition is satisfied:

$$(4.1) \quad \|\xi_{xu}(t) - \xi_{x'u'}(t)\| \leq \beta(\|x - x'\|, t) + \gamma(\|u - u'\|_\infty).$$

A characterization of the above incremental stability notion in terms of dissipation inequalities can be found in [Ang02]. Given a δ -ISS nonlinear control system Σ of the form (2.1), a sampling time $\tau \in \mathbb{R}^+$, a state quantization $\eta \in \mathbb{R}^+$ and an input quantization $\mu \in \mathbb{R}^+$ consider the following system:

$$(4.2) \quad S_{\tau, \eta, \mu}(\Sigma) := (X_{\tau, \eta, \mu}, X_{0, \tau, \eta, \mu}, U_{\tau, \eta, \mu}, \xrightarrow{\tau, \eta, \mu}, Y_{\tau, \eta, \mu}, H_{\tau, \eta, \mu}),$$

where:

- $X_{\tau, \eta, \mu} = [X]_{2\eta}$;
- $X_{0, \tau, \eta, \mu} = X_{\tau, \eta, \mu} \cap X_0$;
- $U_{\tau, \eta, \mu} = [U]_{2\mu}$;
- $x \xrightarrow[\tau, \eta, \mu]{u} y$ if $\xi_{xu}(\tau) \in \mathcal{B}_{[\eta]}(y) \cap X$;
- $Y_{\tau, \eta, \mu} = X$;
- $H_{\tau, \eta, \mu} = \iota : X_{\tau, \eta, \mu} \hookrightarrow Y_{\tau, \eta, \mu}$.

It is readily seen from the definition of $X_{\tau,\eta,\mu}$ and $U_{\tau,\eta,\mu}$ that system $S_{\tau,\eta,\mu}(\Sigma)$ is countable and becomes symbolic when the state space X and the input space U are bounded sets. System $S_{\tau,\eta,\mu}(\Sigma)$ is basically equivalent to the symbolic model proposed in [PGT08]. The main difference is that, while the symbolic model in [PGT08] is not guaranteed to be deterministic, system $S_{\tau,\eta,\mu}(\Sigma)$ is so, as formally stated in the following result:

Proposition 4.2. *System $S_{\tau,\eta,\mu}(\Sigma)$ is deterministic.*

Proof. The existence and uniqueness of a trajectory from an initial condition $x \in X_{\tau,\eta,\mu}$ with input $u \in U_{\tau,\eta,\mu}$ guarantees that $\xi_{xu}(\tau)$ is uniquely determined. Since the collection of sets $\{\mathcal{B}_{[\eta]}(y) \cap X\}_{y \in X_{\tau,\eta,\mu}}$ is a partition of X , there exists at most one state $y \in X_{\tau,\eta,\mu}$ such that $\xi_{xu}(\tau) \in \mathcal{B}_{[\eta]}(y) \cap X$. \square

We stress that determinism in the symbolic system $S_{\tau,\eta,\mu}(\Sigma)$ is an important property because algorithmic synthesis of symbolic systems simplifies when systems are deterministic [Tab09]. We can now give the following result that establishes sufficient conditions for the existence and construction of symbolic systems for nonlinear control systems.

Theorem 4.3. *Consider a δ -ISS nonlinear control system $\Sigma = (X, X_0, U, \mathcal{U}, f)$ and a desired precision $\theta \in \mathbb{R}^+$. For any sampling time $\tau \in \mathbb{R}^+$, state quantization $\eta \in \mathbb{R}^+$ and input quantization $\mu \in \mathbb{R}^+$ satisfying the following inequality:*

$$(4.3) \quad \beta(\theta, \tau) + \gamma(\mu) + \eta \leq \theta,$$

systems $S_{\tau,\eta,\mu}(\Sigma)$ and $S_\tau(\Sigma)$ are θ -approximately bisimilar.

Proof. The proof of the above result can be given along the lines of Theorem 5.1 in [PGT08]. We include it here for the sake of completeness. Consider the relation $\mathcal{R} \subseteq X \times X_{\tau,\eta,\mu}$ defined by $(x, y) \in \mathcal{R}$ if and only if $x \in \mathcal{B}_{[\eta]}(y) \cap X$. We start by showing that condition (i) of Definition 2.6 holds. Consider an initial condition $x_0 \in X_0$. By definition of the set $X_{0,\tau,\eta,\mu}$ there exists $y_0 \in X_{0,\tau,\eta,\mu}$ so that $(x_0, y_0) \in \mathcal{R}$. Condition (ii) in Definition 2.6 is satisfied by the definition of \mathcal{R} . Let us now show that condition (iii) in Definition 2.6 holds. Consider any $(x, y) \in \mathcal{R}$. Consider any $u_1 \in \mathcal{U}_\tau$ and the transition $x \xrightarrow[\tau]{u_1} w$ in $S_\tau(\Sigma)$. There exists $u_2 \in U_{\tau,\eta,\mu}$ such that:

$$(4.4) \quad \|u_2 - u_1\|_\infty \leq \mu.$$

Set $z = \xi_{yu_2}(\tau)$. Since $X = \bigcup_{v \in X_{\tau,\eta,\mu}} \mathcal{B}_{[\eta]}(v) \cap X$, there exists $v \in X_{\tau,\eta,\mu}$ such that:

$$(4.5) \quad z \in \mathcal{B}_{[\eta]}(v),$$

and therefore $y \xrightarrow[\tau,\eta,\mu]{u_2} v$ in $S_{\tau,\eta,\mu}(\Sigma)$. Since Σ is δ -ISS, by the definition of \mathcal{R} and by condition (4.4), the following chain of inequalities holds:

$$\|w - z\| \leq \beta(\|x - y\|, \tau) + \gamma(\|u_1 - u_2\|_\infty) \leq \beta(\theta, \tau) + \gamma(\mu),$$

which implies:

$$(4.6) \quad w \in \mathcal{B}_{\beta(\theta,\tau)+\gamma(\mu)}(z).$$

By combining the inclusions in (4.5) and (4.6), it is readily seen that $w \in \mathcal{B}_{[\beta(\theta,\tau)+\gamma(\mu)+\eta]}(v)$. By the inequality in (4.3), $\mathcal{B}_{[\beta(\theta,\tau)+\gamma(\mu)+\eta]}(v) \subseteq \mathcal{B}_{[\eta]}(v)$, which implies $(w, v) \in \mathcal{R}$ and hence, condition (iii) in Definition 2.6 holds. Thus, condition (i) in Definition 2.7 is satisfied. By using similar arguments it is possible to show condition (ii) of Definition 2.7. \square

The above result is conceptually equivalent to Theorem 5.1 in [PGT08]. The main difference is that while Theorem 4.3 relates nonlinear systems to *deterministic* symbolic systems, Theorem 5.1 in [PGT08] relates nonlinear systems to symbolic models which are in general nondeterministic.

The above result is now employed to define symbolic systems for the plant and the specification. Consider

a plant system P as defined in (3.2) and a specification system Q as defined in (3.3). Suppose that P and Q are δ -ISS and choose a precision $\theta_p \in \mathbb{R}^+$ and a precision $\theta_q \in \mathbb{R}^+$, required in the construction of the symbolic systems for P and Q , respectively. Let β_p and γ_p be a \mathcal{KL} function and a \mathcal{K}_∞ function guaranteeing the δ -ISS stability property for P and β_q be a \mathcal{KL} function guaranteeing the δ -ISS stability property for Q . Find quantization parameters $\tau, \eta, \mu \in \mathbb{R}^+$ such that:

$$(4.7) \quad \begin{aligned} \beta_p(\theta_p, \tau) + \gamma_p(\mu) + \eta &\leq \theta_p, \\ \beta_q(\theta_q, \tau) + \eta &\leq \theta_q, \end{aligned}$$

It is readily seen that parameters $\tau, \eta, \mu \in \mathbb{R}^+$ satisfying the above inequalities always exist. By Theorem 4.3, $S_{\tau, \eta, \mu}(P)$ is θ_p -approximately bisimilar to $S_\tau(P)$ and $S_{\tau, \eta, 0}(Q)$ is θ_q -approximately bisimilar to $S_\tau(Q)$. For the sake of notational simplicity in the further developments we refer to the systems $S_{\tau, \eta, \mu}(P)$ and $S_{\tau, \eta, 0}(Q)$, by means of S_p and S_q , respectively.

4.2. Control Design at the Symbolic Layer. Problem 3.2 translates to the following problem at the symbolic layer:

Problem 4.4. *Given system S_p and system S_q , find a symbolic controller $C \in \mathcal{C}^{\tau, \theta, \mu}(P)$ such that:*

- (i) $(S_p \parallel_0 C) \preceq_0 S_q$;
- (ii) $S_p \parallel_0 C$ is non-blocking.

We start by introducing a technical lemma that will be used in the sequel.

Lemma 4.5. *Consider three metric systems $S_i = (X_i, X_{0,i}, U_i, \xrightarrow{i}, Y, H_i)$, $i = 1, 2, 3$. The following properties hold:*

- (i) [GP07] *For all $\varepsilon_1 \in \mathbb{R}_0^+$, if $S_1 \preceq_{\varepsilon_1} S_2$ then $S_1 \preceq_{\varepsilon_2} S_2$, for all $\varepsilon_2 \geq \varepsilon_1$;*
- (ii) [GP07] *For all $\varepsilon_1, \varepsilon_2 \in \mathbb{R}_0^+$, if $S_1 \preceq_{\varepsilon_1} S_2$ and $S_2 \preceq_{\varepsilon_2} S_3$, then $S_1 \preceq_{\varepsilon_1 + \varepsilon_2} S_3$;*
- (iii) *For all $\varepsilon \in \mathbb{R}_0^+$, $S_1 \parallel_\varepsilon S_2 \preceq_\varepsilon S_2$.*

Proof of (iii). Denote $S_1 \parallel_\varepsilon S_2$ by the tuple $(X, X_0, U, \xrightarrow{\quad}, Y, H)$ and define:

$$\mathcal{R} = \{((x_1, x_2), x) \in X \times X_2 : x_2 = x\}.$$

We start by showing that condition (i) in Definition 2.6 holds. Consider any initial condition $(x_{0,1}, x_{0,2}) \in X_0$. Since $x_{0,2} \in X_2$, by choosing $x_0 = x_{0,2}$ we have that $((x_{0,1}, x_{0,2}), x_0) \in \mathcal{R}$. We now show that also condition (ii) in Definition 2.6 holds. Consider any $((x_1, x_2), x) \in \mathcal{R}$. Since $x_2 = x$, then $H_2(x_2) = H_2(x)$, hence by Definition 2.8 of approximate composition $d(H(x_1, x_2), H_2(x)) = d(H_1(x_1), H_2(x_2)) \leq \varepsilon$. We conclude by showing that condition (iii) in Definition 2.6 holds. Consider any $((x_1, x_2), x) \in \mathcal{R}$ and any transition $(x_1, x_2) \xrightarrow{(u_1, u_2)} (x'_1, x'_2)$ in $S_1 \parallel_\varepsilon S_2$. Choose the transition $x \xrightarrow{\frac{u_2}{2}} x'$ in S_2 so that $x' = x'_2$. By definition of the systems involved such transition exists. This implies that $((x'_1, x'_2), x') \in \mathcal{R}$, which concludes the proof. \square

We are now ready to provide the solution to Problem 4.4. Define:

$$(4.8) \quad C^* = S_p \parallel_0 S_q.$$

Theorem 4.6. *$Nb(C^*)$ solves Problem 4.4.*

Proof. We start by proving condition (i) of Problem 4.4. By Lemma 4.5 (iii), we obtain:

$$(4.9) \quad S_p \parallel_0 Nb(C^*) \preceq_0 Nb(C^*).$$

By the definition of $Nb(C^*)$ it is readily seen that:

$$(4.10) \quad Nb(C^*) \preceq_0 C^*.$$

By the definition of $C^* = S_p \parallel_0 S_q$ and Lemma 4.5 (iii), one gets:

$$(4.11) \quad C^* \preceq_0 S_q.$$

By combining conditions in (4.9), (4.10), (4.11) and by Lemma 4.5 (ii) we obtain:

$$S_p \parallel_0 Nb(C^*) \preceq_0 S_q.$$

Hence, condition (i) of Problem 4.4 is proved. We now prove condition (ii) of Problem 4.4. Consider any state (p_1, p_2, q) of $S_p \parallel_0 Nb(C^*)$. Since $Nb(C^*)$ is non-blocking there exists a state (p_2^+, q^+) of $Nb(C^*)$ so that $(p_2, q) \xrightarrow{u} (p_2^+, q^+)$ is a transition of $Nb(C^*)$ for some input $u = (u_2, u_3)$. Since $Nb(C^*)$ is a sub-system of $C^* = S_p \parallel_0 S_q$, then by choosing $p_1^+ = p_2^+$ and $u_1 = u_2$, the transition $p_1 \xrightarrow{u_1} p_1^+$ is a transition of S_p . Since by construction $p_1^+ = p_2^+$ then (p_1^+, p_2^+, q^+) is a state of $S_p \parallel_0 Nb(C^*)$ and therefore $(p_1, p_2, q) \xrightarrow{(u_1, u)} (p_1^+, p_2^+, q^+)$ is a transition of $S_p \parallel_0 Nb(C^*)$, which concludes the proof. \square

We conclude this section by showing that the controller $Nb(C^*)$ is the maximal system solving Problem 4.4 in the sense of the preorder naturally induced by the notion of 0-simulation relations.

Theorem 4.7. *For any system C solving Problem 4.4*

$$(S_p \parallel_0 C) \preceq_0 (S_p \parallel_0 Nb(C^*)).$$

Proof. Denote by S_p^1 and S_p^2 copies of S_p that are connected to C and $Nb(C^*)$, respectively; denote by X_{pc} and X_{pc^*} the state spaces of $S_p^1 \parallel_0 C$ and $S_p^2 \parallel_0 Nb(C^*)$ and by X_{pc}^0 and $X_{pc^*}^0$ the corresponding sets of initial states. Moreover let $C^* = S_p^c \parallel_0 S_q^c$, where S_p^c and S_q^c are the copies of S_p and S_q in the controller and define:

$$\mathcal{R} = \{((p_1, c), (p_2, p_3, q)) \in X_{pc} \times X_{pc^*} : ((p_1, c), q) \in \mathcal{R}_1 \wedge p_1 = p_2\},$$

where \mathcal{R}_1 is a 0-simulation relation from $S_p^1 \parallel_0 C$ to S_q . We start by showing that condition (i) in Definition 2.6 holds. Consider any initial condition $(p_1^0, c^0) \in X_{pc}^0$. Since $(S_p \parallel_0 C) \preceq_0 S_q$ there exists $q^0 \in S_q$ s.t. $((p_1^0, c^0), q^0) \in \mathcal{R}_1$. By choosing $p_2^0 = p_3^0 = p_1^0$, we have $(p_2^0, p_3^0, q^0) \in X_{pc^*}^0$ and hence, $((p_1^0, c^0), (p_2^0, p_3^0, q^0)) \in \mathcal{R}$. We now show that also condition (ii) in Definition 2.6 holds. Since $H_p(p_1) = H_p(p_2)$, we can conclude $d(H_{pc}(p_1, c), H_{ppq}(p_2, p_3, q)) = d(H_p(p_1), H_p(p_2)) = 0$. We conclude by showing that condition (iii) in Definition 2.6 holds. Consider any $((p_1, c), (p_2, p_3, q)) \in \mathcal{R}$ and any transition $(p_1, c) \xrightarrow{(u_p, u_c)} (p_1^+, c^+)$ in $S_p^1 \parallel_0 C$. Since $((p_1, c), q) \in \mathcal{R}_1$, there exists a transition $q \xrightarrow{v} q^+$ in S_q so that $((p_1^+, c^+), q^+) \in \mathcal{R}_1$. Hence $H_{pc}(p_1^+, c^+) = H_p(p_1^+) = H_q(q^+)$ and $q^+ = p_1^+$. Now, since $p_1 = p_2 = p_3 = q$, we consider the transitions $p_2 \xrightarrow{u_p} p_2^+$ in S_p^2 , $p_3 \xrightarrow{u_p} p_3^+$ in S_p^c and $q \xrightarrow{v} q^+$ in S_q^c with $p_3^+ = p_2^+ = p_1^+ = q^+$. Notice that such transitions exist. Hence (p_2^+, p_3^+, q^+) is a state of $S_p^2 \parallel_0 Nb(C^*)$ and the transition $(p_2, p_3, q) \xrightarrow{(u_p, u_p, v)} (p_2^+, p_3^+, q^+)$ is in $S_p^2 \parallel_0 Nb(C^*)$, which implies $((p_1^+, c^+), (p_2^+, p_3^+, q^+)) \in \mathcal{R}$. \square

This result is important because it shows that the controller $Nb(C^*)$ implements the maximal non-blocking behaviour of the specification symbolic system S_q , which can be implemented by the plant symbolic system S_p .

4.3. From the Symbolic Layer to the Continuous Layer. We now have all the ingredients to present one of the main results of this paper which shows that there exists an appropriate choice of quantization parameters so that the symbolic controller $Nb(C^*)$ with C^* defined in (4.8) solves Problem 3.2.

Theorem 4.8. *Consider the plant system P as in (3.2), the specification system Q as in (3.3) and a precision $\varepsilon \in \mathbb{R}^+$. Suppose that P and Q are δ -ISS and choose parameters $\theta_p, \theta_q \in \mathbb{R}^+$ so that:*

$$(4.12) \quad \theta_p + \theta_q \leq \varepsilon.$$

Furthermore choose parameters $\tau, \eta, \mu \in \mathbb{R}^+$ satisfying the inequalities in (4.7). Then the symbolic controller $Nb(C^*) \in C^{\tau, \theta, \mu}(P)$ with $\theta = \theta_p$ and C^* defined in (4.8) with $S_p = S_{\tau, \eta, \mu}(P)$ and $S_q = S_{\tau, \eta, 0}(Q)$, solves Problem 3.2.

Proof. We start by proving condition (i) of Problem 3.2. By Lemma 4.5 (iii), we obtain:

$$(4.13) \quad S_\tau(P) \parallel_{\theta_p} Nb(C^*) \preceq_{\theta_p} Nb(C^*).$$

By the definition of $Nb(C^*)$ it is readily seen that:

$$(4.14) \quad Nb(C^*) \preceq_0 C^*.$$

By the definition of $C^* = S_p \parallel_0 S_q$ and Lemma 4.5 (iii), one gets:

$$(4.15) \quad C^* \preceq_0 S_q.$$

Since S_q is θ_q -approximately bisimilar to $S_\tau(Q)$ then:

$$(4.16) \quad S_q \preceq_{\theta_q} S_\tau(Q).$$

By combining conditions in (4.13), (4.14), (4.15), (4.16) and by Lemma 4.5 (ii) we obtain:

$$S_\tau(P) \parallel_{\theta_p} Nb(C^*) \preceq_{\theta_p + \theta_q} S_\tau(Q).$$

Since by (4.12), $\theta_p + \theta_q \leq \varepsilon$, by Lemma 4.5 (i), condition (i) of Problem 3.2 is proved. We now prove condition (ii) of Problem 3.2. Consider any state (p_1, p_2, q) of $S_\tau(P) \parallel_{\theta_p} Nb(C^*)$. Since $Nb(C^*)$ is non-blocking there exists a state (p_2^+, q^+) of $Nb(C^*)$ so that $(p_2, q) \xrightarrow{u} (p_2^+, q^+)$ is a transition of $Nb(C^*)$ for some input $u = (u_2, u_3)$. Since $S_\tau(P)$ and S_p are θ_p -approximately bisimilar, for the transition $p_2 \xrightarrow{u_2} p_2^+$ in S_p there exists a transition $p_1 \xrightarrow{u_1} p_1^+$ in $S_\tau(P)$ so that $d(H_p(p_1^+), H_p(p_2^+)) \leq \theta_p$. This implies that (p_1^+, p_2^+, q^+) is a state of $S_\tau(P) \parallel_{\theta_p} Nb(C^*)$ and therefore that $(p_1, p_2, q) \xrightarrow{(u_1, u)} (p_1^+, p_2^+, q^+)$ is a transition of $S_\tau(P) \parallel_{\theta_p} Nb(C^*)$, which concludes the proof. \square

5. INTEGRATED SYMBOLIC CONTROL DESIGN

The construction of the symbolic controller $Nb(S_p \parallel_0 S_q)$ solving Problem 3.2 relies upon the basic-steps procedure illustrated in Algorithm 1.

- 1 Construct system S_p , θ_p -approximately bisimilar to $S_\tau(P)$;
- 2 Construct system S_q , θ_q -approximately bisimilar to $S_\tau(Q)$;
- 3 Construct the composition $S_p \parallel_0 S_q$;
- 4 Compute the non-blocking part $Nb(S_p \parallel_0 S_q)$ of $S_p \parallel_0 S_q$.

Algorithm 1: Construction of $Nb(S_p \parallel_0 S_q)$.

The procedure in Algorithm 1 is common with other approaches currently available in the literature for symbolic control design of continuous and hybrid systems, see e.g. [TP06, YB09, Tab08]. Software implementation of Algorithm 1 requires that:

- State space X_p and set of input values U_p of P are bounded;
- State space X_q of Q is bounded.

The above assumptions, while being reasonable in many realistic engineering control problems, are also needed to store the transitions of systems S_p and S_q in a computer machine, whose memory resources are limited by their nature. In this section, we suppose that the plant P and the specification Q satisfy the above assumptions. The procedure illustrated in Algorithm 1 is not efficient from the space and time complexity point of view³ because:

³This qualitative claim will be substantiated in terms of complexity analysis in the next section.

- It considers the whole state spaces of the plant P and the specification Q . A more efficient algorithm would consider only the intersection of the accessible parts of P and Q .
- For any source state x and target state y it includes all transitions (x, u, y) with any control input u by which state x reaches state y . A more efficient algorithm would consider for any source state x and target state y only one control input u and hence, only one transition.
- It first construct the symbolic models S_p and S_q , then the composed system $S_p \parallel_0 S_q$ to finally eliminate blocking states from $S_p \parallel_0 S_q$. A more efficient algorithm would eliminate blocking states as soon as they show up.

Inspired from the research line in the context of on-the-fly verification and control of timed or untimed transition systems (see e.g. [CVWY92, TA99]), we now present an algorithm which *integrates each step of the four sub-algorithms in Algorithm 1 in only one algorithm*.

The proposed procedure is composed of Algorithm 2 and Algorithm 3. Algorithm 2 is the main one while Algorithm 3 introduces Function **NonBlock**, which is recursively used in Algorithm 2. The outcome of Algorithm 2 is the symbolic controller C^{**} which will be shown in the further results to solve Problem 3.2. Given a set $T \subseteq X \times U \times Y$, the set $\mathbf{X}_{source}(T) \subseteq X$ denotes the projection of T onto X , i.e.

$$\mathbf{X}_{source}(T) = \{x \in X : \exists y \in Y \wedge \exists u \in U \text{ s.t. } (x, u, y) \in T\}.$$

Given a vector $x \in \mathbb{R}^n$ and a precision $\eta \in \mathbb{R}^+$, the symbol $[x]_{2\eta}$ denotes the unique vector in $[\mathbb{R}^n]_{2\eta}$ such that $x \in \mathcal{B}_{[\eta]}([x]_{2\eta})$. Algorithm 2 proceeds as follows. The set of states X_0 of C^{**} is initialized to be $[X_{p,0} \cap X_{q,0}]_{2\eta}$ in line 2.8 and the set of states to be processed, denoted by \mathbf{X}_{target} , is initialized to the set of initial states in line 2.9. The set T of transitions and the set Bad of blocking states of C^{**} are initialized to be the empty-sets (lines 2.10, 2.11). At each basic step, Algorithm 2 processes a (non processed) state in line 2.13, by computing the state $y = [\xi_x^q(\tau)]_{2\eta}$ (line 2.14). If the state y is non-blocking (line 2.15), the algorithm looks for a control input $u \in [U]_{2\mu}$ such that the plant P meets the specification Q , i.e. $z = y$ (line 2.20). If such a control input u exists, then boolean variable $Flag$ is updated to 1 (line 2.21), the transition (x, u, y) is added to the set of transitions T (line 2.25), and the state y is added to the set of the to-be-processed states (line 2.26). If either state y is blocking or no inputs are found for the plant P to meet the specification Q , then state x is declared blocking, and Function **NonBlock**(T, x, Bad) in Algorithm 3 is invoked (line 2.30), in order to remove all blocking states originating from x . Algorithm 2 proceeds with further basic steps, until there are no more states to be processed. When Algorithm 2 terminates, it returns in line 2.34 the symbolic controller C^{**} . Function **NonBlock**(T, x, Bad) extracts the non-blocking part of T . The set $Badx$ includes the states to be processed and is initialized to contain the only state x (line 3.3). At each basic step, for any $y \in Badx$, Function **NonBlock** removes from the set T any transition (z, u, y) ending up in y (line 3.7), it adds z to the set $Badx$ of states to be processed (line 3.8) and adds y to the set Bad of blocking states (lines 3.11, 3.12). Function **NonBlock** terminates when there are no more states to be processed and returns in line 2.14 the updated sets of transitions T of and blocking states Bad . Termination of Algorithm 2 is discussed in the following result:

Theorem 5.1. *Algorithm 2 terminates in a finite number of steps.*

Proof. Algorithm 2 terminates when there are no more states x in \mathbf{X}_{target} to be processed. For each state x , either line 2.25 or line 2.30 is executed (depending on the value of the boolean variable $Flag$); this ensures by line 2.13 that state x cannot be processed again in future iterations. Furthermore, the set \mathbf{X}_{target} is nondecreasing (see line 2.26) and always contained in the finite set $[X_p]_{2\eta} \cap [X_q]_{2\eta}$. Hence, provided that Algorithm 3 terminates in finite time, the result follows. Regarding termination of Algorithm 3, in the worst case the set Bad ends up in coinciding with the accessible states of S_p and S_q (line 3.12) and the set $Badx$ ends up in being empty (line 3.11). Hence from line 3.4, finite termination of Algorithm 3 is guaranteed. \square

Formal correctness of Algorithm 2 is guaranteed by the following result.

Theorem 5.2. *Controllers $Nb(C^*)$ and C^{**} are exactly bisimilar.*

```

1 Input:
2 Plant:  $P = (X_p, X_{p,0}, U_p, \mathcal{U}_p, f_p)$ ;
3 Specification:  $Q = (X_q, X_{q,0}, U_q, \mathcal{U}_q, f_q)$ ;
4 Precision:  $\varepsilon \in \mathbb{R}^+$ ;
5 Parameters:  $\theta_p, \theta_q \in \mathbb{R}^+$  satisfying (4.12);
6 Parameters:  $\tau, \eta, \mu \in \mathbb{R}^+$  satisfying (4.7);
7 Init:
8  $X_0 := [X_{p,0} \cap X_{q,0}]_{2\eta}$ ;
9  $\mathbf{X}_{target} = X_0$ ;
10  $T := \emptyset$ ;
11  $Bad := \emptyset$ ;
12 foreach  $x \in [X_p \cap X_q]_{2\eta}$  do
13     if  $x \in \mathbf{X}_{target} \setminus (\mathbf{X}_{source}(T) \cup Bad)$  then
14         compute  $y = [\xi_x^q(\tau)]_{2\eta}$ ;
15         if  $y \notin Bad$  then
16              $Flag := 0$ ;
17             while  $Flag = 0$  do
18                 choose  $u \in [U_p]_{2\mu}$ ;
19                 compute  $z = [\xi_{xu}^p(\tau)]_{2\eta}$ ;
20                 if  $z = y$  then
21                      $Flag := 1$ ;
22                 end
23             end
24             if  $Flag = 1$  then
25                  $T := T \cup \{(x, u, y)\}$ ;
26                  $\mathbf{X}_{target} := \mathbf{X}_{target} \cup \{y\}$ ;
27             end
28         end
29         if  $Flag = 0 \vee y \in Bad$  then
30              $(T, Bad) := \text{NonBlock}(T, x, Bad)$ ;
31         end
32     end
33 end
34 output:  $C^{**} = (\mathbf{X}_{source}(T), X_0 \cap \mathbf{X}_{source}(T), [U_p]_{2\mu}, T, Y_{\tau, \eta, \mu}, H_{\tau, \eta, \mu})$ 
    
```

Algorithm 2: Integrated Symbolic Control Design.

Proof. (Sketch.) For any state (x_p, x_q) of the accessible part $Ac(Nb(C^*))$ of $Nb(C^*)$ there exists a state x_c of C^{**} so that $x_p = x_q = x_c$ (see lines 2.14, 2.19, 2.20 and 2.25 in Algorithm 2). Consider the relation defined by $((x_p, x_q), x_c) \in \mathcal{R}$ if and only if $x_p = x_c$. It is readily seen that \mathcal{R} is a 0-bisimulation relation between $Nb(C^*)$ and C^{**} . \square

By the above result the controller $Nb(C^*)$ solves Problem 3.2 if and only if the controller C^{**} solves Problem 3.2. Hence, it shows that Algorithm 2 is correct. While the controllers $Nb(C^*)$ and C^{**} are exactly bisimilar, the number of states of C^{**} is in general, smaller than the one of $Nb(C^*)$. In fact the controller $Nb(C^*)$ may contain spurious states, e.g. states which are not accessible from a quantized initial condition in S_p and a quantized initial condition in S_q , since in general $Ac(Nb(C^*))$ is a (strict) sub-system of $Nb(C^*)$. On the other hand, a straightforward inspection of Algorithm 2 reveals that:

```

1 Function  $(T, Bad) := \text{NonBlock}(T, x, Bad)$ ;
2 Init:
3  $Badx := \{x\}$ ;
4 foreach  $y \in Badx$  do
5   foreach  $z \in \mathbf{X}_{source}(T)$  do
6     if  $\exists u \in [U]_{2\mu}$  such that  $(z, u, y) \in T$  then
7        $T := T \setminus \{(z, u, y)\}$ ;
8        $Badx := Badx \cup \{z\}$ ;
9     end
10  end
11   $Badx := Badx \setminus \{y\}$ ;
12   $Bad := Bad \cup \{y\}$ ;
13 end
14 output:  $(T, Bad)$ 

```

Algorithm 3: Non-blocking Algorithm.

Proposition 5.3. $Ac(C^{**}) = C^{**}$.

Hence, the aforementioned spurious states of $Nb(C^*)$ are not included in C^{**} . The above remarks suggest the following result:

Theorem 5.4. C^{**} is the minimal 0-bisimilar system of $Nb(C^*)$.

Proof. The proof can be given by using standard arguments on bisimulation theory [CGP99]. Briefly, since by Proposition 5.3 $Ac(C^{**}) = C^{**}$ and since the output function $H_{\tau, \eta, \mu}$ of C^{**} is the natural inclusion from $\mathbf{X}_{source}(T)$ to X , the maximal 0-bisimulation relation \mathcal{R}^* between C^{**} and itself is the identity relation, i.e. $\mathcal{R}^* = \{(x_1, x_2) \in \mathbf{X}_{source}(T) \times \mathbf{X}_{source}(T) : x_1 = x_2\}$. Since \mathcal{R}^* is the identity relation, the quotient of C^{**} induced by \mathcal{R}^* , coincides with C^{**} . Finally, since by Theorem 5.2 systems C^{**} and $Nb(C^*)$ are 0-bisimilar, the result follows. \square

The above result is important because it shows that the controller C^{**} is the system with the smallest number of states which is equivalent by bisimulation to the solution $Nb(C^*)$ of Problem 3.2.

6. SPACE AND TIME COMPLEXITY ANALYSIS

In this section we provide a formal comparison in terms of space and time complexity analysis, between the procedure illustrated in Algorithm 1 and Algorithm 2.

Proposition 6.1. *Space complexity of Algorithm 1 is $O(\max\{\text{card}([X_p]_{2\eta}) \cdot \text{card}([U_p]_{2\mu}), \text{card}([X_q]_{2\eta})\})$.*

Proof. Since by Proposition 4.2 system S_p is deterministic, the number of transitions of S_p amounts to $\text{card}([X_p]_{2\eta}) \cdot \text{card}([U_p]_{2\mu})$. For the same reason, the number of transitions of S_q is given by $\text{card}([X_q]_{2\eta})$. By definition of exact composition (see Definition 2.8 with $\varepsilon = 0$), the number of transitions in $S_p \parallel_0 S_q$ amounts in the worst case to $(\text{card}([X_p]_{2\eta}) \cap \text{card}([X_q]_{2\eta})) \cdot \text{card}([U_p]_{2\mu})$. By definition of the Nb operator, the number of transitions in $Nb(C^*)$ is less than or equal to the one of $S_p \parallel_0 S_q$. Hence, by comparing the above worst case bounds, the result follows. \square

Proposition 6.2. *Space complexity of Algorithm 2 is $O(\text{card}([X_p]_{2\eta} \cap [X_q]_{2\eta}))$.*

Proof. By lines 2.14, 2.15, 2.20, and 2.25 in Algorithm 2, the triple (x, u, y) is added to the set T of transitions of C^{**} , if (x, u, y) is a transition of S_p and (x, y) is a transition of S_q . Hence, the result follows from determinism of systems S_p and S_q , which is guaranteed by Proposition 4.2. \square

By comparing Propositions 6.1 and 6.2, it is readily seen that space complexity of Algorithm 2 is smaller than or equal to space complexity of Algorithm 1. In particular, when the plant system P and the specification system Q coincide, implying $[X_p]_{2\eta} = [X_q]_{2\eta}$ and $\text{card}([U_p]_{2\mu}) = 1$, the space complexity of the procedure in Algorithm 1 and of Algorithm 2 coincides, resulting in $O(\text{card}([X_p]_{2\eta})) = O(\text{card}([X_q]_{2\eta}))$. This is indeed consistent with the integration philosophy that we advocated in Algorithm 2. Algorithm 2 becomes more and more efficient from the space complexity point of view as much as the behaviours of the plant and of the specification differ. When P and Q coincide there is no gain in terms of space complexity, in the use of Algorithm 2. We now proceed with a further step by providing a comparison in terms of time complexity analysis.

Proposition 6.3. *Time complexity of Algorithm 1 is $O(\text{card}([X_q]_{2\eta}) \cdot \text{card}([X_p]_{2\eta}) \cdot \text{card}([U_p]_{2\mu}))$.*

Proof. The number of steps needed in the construction of S_p and S_q amounts to $\text{card}([X_p]_{2\eta}) \cdot \text{card}([U_p]_{2\mu})$ and $\text{card}([X_q]_{2\eta})$, respectively. Since as shown in Proposition 6.1 the number of transitions in S_p and S_q is given respectively by $\text{card}([X_p]_{2\eta}) \cdot \text{card}([U_p]_{2\mu})$ and $\text{card}([X_q]_{2\eta})$, the number of steps needed in the construction of $S_p \parallel_0 S_q$ is given by $\text{card}([X_q]_{2\eta}) \cdot \text{card}([X_p]_{2\eta}) \cdot \text{card}([U_p]_{2\mu})$. Regarding the computation of the non-blocking part $Nb(S_p \parallel_0 S_q)$, in the worst case for any state of $S_p \parallel_0 S_q$, i.e. for any state in $[X_q \cap X_p]_{2\eta}$, all transitions in $S_p \parallel_0 S_q$ are needed to be processed in order to find blocking states. Since the number of transitions in $S_p \parallel_0 S_q$ is $\text{card}([X_q \cap X_p]_{2\eta}) \cdot \text{card}([U_p]_{2\mu})$, the overall number of steps needed in the computation of $Nb(S_p \parallel_0 S_q)$ is given by $\text{card}([X_q \cap X_p]_{2\eta})^2 \cdot \text{card}([U_p]_{2\mu})$. By comparing the above worst case bounds, the result follows. \square

Proposition 6.4. *Time complexity of Algorithm 2 is*

$$O(\max\{\text{card}([X_q \cap X_p]_{2\eta}) \cdot \text{card}([U_p]_{2\mu}), \text{card}([X_q \cap X_p]_{2\eta})^2\}).$$

Proof. By exploring Algorithm 2, it is easy to see that the number of steps needed in the computation of C^{**} is upper bounded by:

$$(6.1) \quad \sum_{i=0}^{N_1} (N_2 + N_3),$$

where $N_1 = \text{card}([X_p \cap X_q]_{2\eta})$, N_2 is an upper bound to the number of steps needed in the execution of lines 2.13/27 in Algorithm 2, and N_3 is an upper bound to the number of steps needed in the execution of lines 2.28/30 in Algorithm 2. Quantity in (6.1) can be rewritten as the sum of the term $\sum_{i=0}^{N_1} N_2$ and the term $\sum_{i=0}^{N_1} N_3$, the first of which is upper bounded by $\text{card}([X_p \cap X_q]_{2\eta}) \cdot \text{card}([U_p]_{2\mu})$. Regarding the term $\sum_{i=0}^{N_1} N_3$, whenever Algorithm 2 executes line 2.30, i.e. $(T, Bad) := \text{NonBlock}(T, x, Bad)$, states x involved are different. Indeed suppose by contradiction that at step i state x is processed in line 2.30 and at step j state x' is processed in line 2.30 with $i < j$ and $x = x'$. When at step i Algorithm 3 is invoked, state x is added to the set Bad (see lines 3.3, 3.4 and 3.12). Since at the end of step i state $x \in Bad$, in the further steps and in particular at step j , state x will be no longer processed (see line 2.13). Since $x' = x$, then at step j state x' cannot be processed in line 2.30. Hence a contradiction holds. Since any time Algorithm 3 is invoked it processes different states, the overall time complexity due to the term $\sum_{i=0}^{N_1} N_3$ is upper bounded by the time complexity needed in computing the non-blocking part of $S_p \parallel_0 S_q$ which, from Proposition 6.3 amounts to $\text{card}([X_q \cap X_p]_{2\eta})^2 \cdot \text{card}([U_p]_{2\mu})$. By comparing the above worst case bounds, the result follows. \square

By comparing Propositions 6.3 and 6.4, it is readily seen that time complexity of Algorithm 2 is smaller than or equal to time complexity of Algorithm 1. In particular, when the plant system P and the specification

system Q coincide, implying $[X_p]_{2\eta} = [X_q]_{2\eta}$ and $\text{card}([U_p]_{2\mu}) = 1$, the time complexity of the procedure in Algorithm 1 and of Algorithm 2 coincides, resulting in $O(\text{card}([X_p]_{2\eta})^2) = O(\text{card}([X_q]_{2\eta})^2)$.

7. EXAMPLES

In this section we present some examples of application of the results illustrated in the previous sections. In particular, we consider in Section 7.1 symbolic control design problem for a nonlinear control system and in Section 7.2 symbolic control design for linear control systems. The results shown hereafter are based on computations performed on an Intel Core 2 Duo T5500 1.66GHz laptop with 4 GB RAM.

7.1. Nonlinear Control Systems. Consider the following plant nonlinear control system:

$$P : \begin{cases} \dot{x}_1 = -2x_1 + x_3^2 - u \\ \dot{x}_2 = 2x_1 - 7e^{x_2} + 7 \\ \dot{x}_3 = -3x_3 + \frac{3}{4}u^2, \end{cases}$$

and an infinite states specification, expressed by the following differential equation:

$$Q : \begin{cases} \dot{x}_1 = -3x_1 + x_3^3 \\ \dot{x}_2 = x_1 - 5 \sin x_2 \\ \dot{x}_3 = -x_2^2 - 4x_3. \end{cases}$$

We suppose for simplicity that the plant and the specification systems share the same state space, chosen as:

$$X_p = X_q = [-1, 1] \times [-1, 1] \times [-1, 1],$$

the same set of initial states, chosen as:

$$X_p^0 = X_q^0 = [-1, 0] \times [-1, 0] \times [-1, 0],$$

and that the plant input space is:

$$U = [-1, 1].$$

By using the δ -ISS Lyapunov characterization in [Ang02] it is possible to show the plant system P is δ -ISS with functions:

$$\beta_p(r, s) := \sqrt{2} e^{-1.21s} r, \quad \gamma(r) := \sqrt{14.88} r, \quad r, s \in \mathbb{R}_0^+.$$

Analogously the specification system Q can be shown to be δ -ISS with function:

$$\beta_q(r, s) := \sqrt{2} e^{-s} r, \quad r, s \in \mathbb{R}_0^+.$$

For a precision $\varepsilon = 0.2$, we can choose the following quantization parameters for the plant and the specification systems:

$$\theta_p = 0.13, \quad \theta_q = 0.07, \quad \eta = 1/30, \quad \tau = 1, \quad \mu = 0.001.$$

The above choice of quantization parameters guarantees that the inequalities in (4.7) and (4.12) are fulfilled. By running Algorithm 2 the integrated symbolic controller C^{**} has been designed. Given the large size of the controller obtained (3152 states) we do not report in the paper further details on it. Figures 2 shows the evolution of the plant system P when interconnected with the symbolic controller C^{**} and the evolution of the specification system Q , with initial condition $x_0 = (-1, -1, -1 + 4\eta)$. It is readily seen from the plots that for the initial condition x_0 the specification is fulfilled, up to the precision $\varepsilon = 0.2$ chosen in this example.

We conclude this section by discussing a comparison between the “integrated” approach formulated in Algorithm 2 and the “non-integrated” approach described in Algorithm 1. Experimental results associated with the computation of C^{**} and of $Nb(C^*)$ are reported in Tables 1 and 2. In particular, Table 1 shows details in the computation of the controller $Nb(C^*)$ performed by running Algorithm 1. Table 2 reports a comparison between the computation of the controllers C^{**} and $Nb(C^*)$. The computation time needed in the construction of the controllers is expressed in seconds and the maximal memory occupation is given in terms of the maximal number of data needed in the construction of the controllers. In particular, the maximal memory occupation in the construction of $Nb(C^*)$ is expressed as the sum of the number of transitions of S_p , the

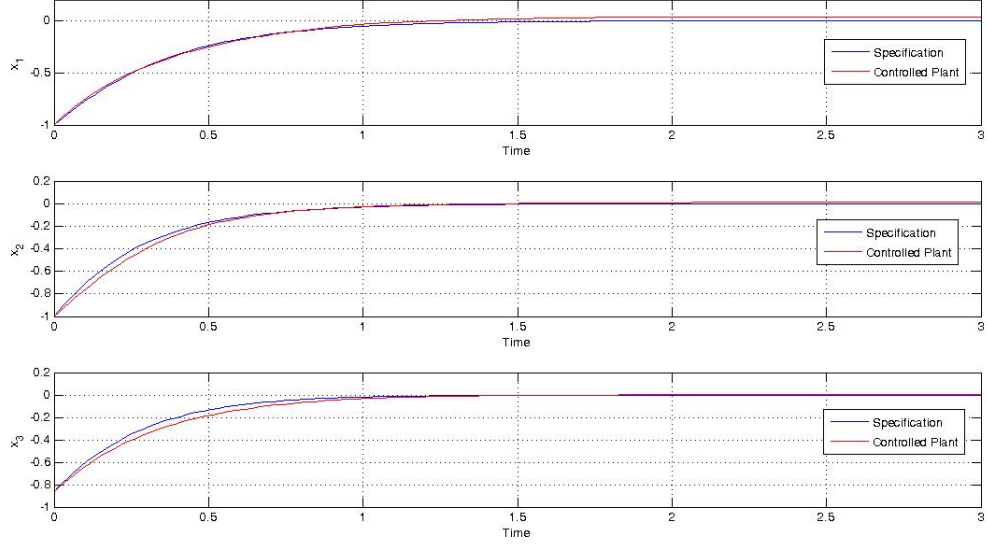


FIGURE 2. Trajectories of the controlled plant and the specification systems with initial condition $(-1, -1, -1 + 4\eta)$.

	S_p	S_q	C^*	$Nb(C^*)$
States	29791	29791	21894	21894
Transitions	29820791	29791	1265217	1265217

TABLE 1. Details on the computation of $Nb(C^*)$.

	$Nb(C^*)$	C^{**}	Ratio
States	21894	3152	0.14
Transitions	1265217	3152	$2.5 \cdot 10^{-3}$
Max memory occupation	93347397	10400	$1.11 \cdot 10^{-4}$
Time	147487	11144	0.08

TABLE 2. Comparison in the computation of $Nb(C^*)$ and C^{**} .

number of transitions of S_q and the number of transitions of $S_p \parallel S_q$, while the maximal memory occupation in the construction of C^{**} is given as the sum of the number of transitions in C^{**} and the number of states in *Bad*. For both controllers $Nb(C^*)$ and C^{**} each transition is weighted as three data and each state as one datum. The experimental results shown in Table 2 can be summarized, as follows:

- The number of states of C^{**} is 14% times the number of states of $Nb(C^*)$;
- The number of transitions of C^{**} is 0.25% times the number of transitions of $Nb(C^*)$;
- The maximal memory occupation of C^{**} is 0.011% times the maximal memory occupation of $Nb(C^*)$;
- The time needed in the computation of C^{**} is 8% times the time of computation of $Nb(C^*)$.

7.2. Linear Control Systems. In this section we consider eight examples randomly chosen in the class of linear systems, characterized by different properties regarding controllability and eigenvalues of dynamical

matrices. We consider controllable, versus noncontrollable plant systems (Examples no. 1, 2, 3, 4 vs. 5, 6, 7, 8), plant dynamical matrices A_p with real, versus complex eigenvalues (Examples no. 3, 4, 5, 6, 7, 8 vs. 1, 2), specification dynamical matrices A_q with real, versus complex eigenvalues (Examples no. 2, 3, 7 vs. 1, 4, 5, 6, 8). Table 3 shows the experimental results. In particular lines 1.1, 1.2 and 1.3 show respectively, dynamical matrices A_p and B_p of the plant P and dynamical matrices A_q of the specification Q . For simplicity we consider in the eight examples the same state space of the plant and the specification, chosen as:

$$X_p = X_q = [-0.5, 0.5[\times [-0.5, 0.5[,$$

the same set of initial states of P and Q , chosen as:

$$X_p^0 = X_q^0 = [-0.25, 0.25[\times [-0.25, 0.25[,$$

and the same input space, chosen as:

$$U = [-2, 2].$$

The quantization parameters in the construction of the symbolic systems S_p and S_q are the same in all the examples and chosen as:

$$\varepsilon = 0.1, \quad \tau = 0.5, \quad \mu = 0.001, \quad \eta = 0.01, \quad \theta_p = 0.05, \quad \theta_q = 0.05.$$

It is readily seen that the above parameters satisfy the inequalities in (4.7) and (4.12). Experimental results associated with the computation of the controller $Nb(C^*)$ are reported in lines 2.1/2.10. In particular, line 2.10 shows the time of computation needed in the construction of $Nb(C^*)$ and line 2.9 shows the maximal memory occupation in the construction of $Nb(C^*)$. Experimental results associated with the computation of the controller C^{**} are reported in lines 3.1/3.5. In particular line 3.5 shows the time of computation needed in the construction of C^{**} and line 3.4 shows the maximal memory occupation in the construction of C^{**} . Table 4 summarizes the results shown in Table 3:

- **Line 4.1: Gain in terms of number of states.** The minimum gain of the integrated procedure versus the non-integrated procedure is obtained in Example # 5, resulting in 100% (meaning that in this example there is no gain in the integrated procedure) and, the maximum gain is obtained in Example # 7, resulting in 53%.
- **Line 4.2: Gain in terms of number of transitions.** The minimum gain of the integrated procedure versus the non-integrated procedure is obtained in Example # 3, resulting in 5% and, the maximum gain is obtained in Example # 7, resulting in 2%.
- **Line 4.3: Gain in terms of maximal memory occupation.** The minimum gain of the integrated procedure versus the non-integrated procedure is obtained in Examples # 2 and 4, resulting in 0.017% and, the maximum gain is obtained in Example # 8, resulting in 0.007%.
- **Line 4.4: Gain in terms of time of computation.** The minimum gain of the integrated procedure versus the non-integrated procedure is obtained in Example # 3, resulting in 28% and, the maximum gain is obtained in Example # 7, resulting in 9%.

8. DISCUSSION

In this paper we addressed the problem of symbolic control design of nonlinear systems with infinite states specifications, modelled by differential equations. After having provided an explicit solution to the symbolic control design problem, we presented Algorithm 2 which integrates the design of the symbolic controller with the construction of the symbolic systems of the plant and of the specification. Although the focus of the present paper is on infinite states specifications, it can be shown that the results here presented can be easily adapted to consider finite states specifications which include language specifications, formalized through automata theory [CL99]. This is important because, as shown in the work of [TP06, Tab08, BH06], automata theory provides a novel class of specifications which were traditionally not addressed before, in the control design of continuous (nonlinear) systems. Future work will focus on more efficient techniques at the software layer which can further reduce space and time complexity in the implementation of Algorithm 2. Useful insights in

	Example # 1	Example # 2	Example # 3	Example # 4
1. Data				
1.1 A_p	$\begin{pmatrix} -1 & -0.5 \\ 0.5 & -1 \end{pmatrix}$	$\begin{pmatrix} -0.8 & -0.3 \\ 0.3 & -0.8 \end{pmatrix}$	$\begin{pmatrix} -0.6 & 0 \\ 0 & -0.1 \end{pmatrix}$	$\begin{pmatrix} -0.9 & 0 \\ 0 & -0.6 \end{pmatrix}$
1.2 B_p	$\begin{pmatrix} 1 & 1 \end{pmatrix}'$	$\begin{pmatrix} 1 & 1 \end{pmatrix}'$	$\begin{pmatrix} 1 & 1 \end{pmatrix}'$	$\begin{pmatrix} 1 & 1 \end{pmatrix}'$
1.3 A_q	$\begin{pmatrix} -0.75 & -0.25 \\ 0.25 & -0.75 \end{pmatrix}$	$\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix}$	$\begin{pmatrix} -1 & 0 \\ 0 & -2 \end{pmatrix}$	$\begin{pmatrix} -0.8 & -0.4 \\ 0.4 & -0.8 \end{pmatrix}$
2. Nb(C^*)				
2.1 States of S_p	2601	2601	2601	2601
2.2 Transitions of S_p	2675069	2494785	2489327	2446901
2.3 States of S_q	2601	2601	2601	2601
2.4 Transitions of S_q	2601	2601	2601	2601
2.5 States of C^*	611	603	403	915
2.6 Transitions of C^*	8013	7507	4969	11151
2.7 States of $Nb(C^*)$	403	521	343	499
2.8 Transitions of $Nb(C^*)$	5719	6753	4331	6505
2.9 Max($Nb(C^*)$)	8057049	7514679	7490691	7381959
2.10 Time($Nb(C^*)$)	7780	7095	4648	4068
3. C^{**}				
3.1 States of C^{**}	239	281	199	277
3.2 Transitions of C^{**}	239	281	199	277
3.3 States in <i>Bad</i>	490	448	530	452
3.4 Max(C^{**})	1207	1291	1127	1283
3.5 Time(C^{**})	1300	1800	1300	770
	Example # 5	Example # 6	Example # 7	Example # 8
1. Data				
1.1 A_p	$\begin{pmatrix} -0.9 & 0 \\ 0 & -0.6 \end{pmatrix}$	$\begin{pmatrix} -1.5 & 1 \\ 0 & -1.5 \end{pmatrix}$	$\begin{pmatrix} -0.8 & 0 \\ 0 & -0.6 \end{pmatrix}$	$\begin{pmatrix} -1.5 & 1 \\ 0 & -1.5 \end{pmatrix}$
1.2 B_p	$\begin{pmatrix} 1 & 0 \end{pmatrix}'$	$\begin{pmatrix} 1 & 0 \end{pmatrix}'$	$\begin{pmatrix} 1 & 0 \end{pmatrix}'$	$\begin{pmatrix} 1 & 0 \end{pmatrix}'$
1.3 A_q	$\begin{pmatrix} -0.8 & -0.4 \\ 0.4 & 0.8 \end{pmatrix}$	$\begin{pmatrix} -0.9 & -0.5 \\ 0.5 & -0.9 \end{pmatrix}$	$\begin{pmatrix} -1.2 & 0 \\ 0 & -0.5 \end{pmatrix}$	$\begin{pmatrix} -1 & -0.75 \\ 0.75 & -1 \end{pmatrix}$
2. Nb(C^*)				
2.1 States of S_p	2601	2601	2601	2601
2.2 Transitions of S_p	3290367	3721269	3215397	3721269
2.3 States of S_q	2601	2601	2601	2601
2.4 Transitions of S_q	2601	2601	2601	2601
2.5 States of C^*	381	325	1377	227
2.6 Transitions of C^*	9467	9233	33453	6451
2.7 States of $Nb(C^*)$	99	129	153	65
2.8 Transitions of $Nb(C^*)$	2461	3665	3717	1847
2.9 Max($Nb(C^*)$)	9907305	11199309	9754353	11190963
2.10 Time($Nb(C^*)$)	6285	7080	4880	9444
3. C^{**}				
3.1 States of C^{**}	99	109	81	53
3.2 Transitions of C^{**}	99	109	81	53
3.3 States in <i>Bad</i>	630	620	648	676
3.4 Max(C^{**})	927	947	891	835
3.5 Time(C^{**})	920	850	430	990

 TABLE 3. Details on the computation of $Nb(C^*)$ and C^{**} .

	Example # 1	Example # 2	Example # 3	Example # 4
4.1 $\text{States}(C^{**})/\text{States}(Nb(C^*))$	0.59	0.54	0.58	0.55
4.2 $\text{Transitions}(C^{**})/\text{Transitions}(Nb(C^*))$	0.04	0.04	0.05	0.04
4.3 $\text{Max}(C^{**})/\text{Max}(Nb(C^*))$	$1.5 \cdot 10^{-4}$	$1.7 \cdot 10^{-4}$	$1.5 \cdot 10^{-4}$	$1.7 \cdot 10^{-4}$
4.4 $\text{Time}(C^{**})/\text{Time}(Nb(C^*))$	0.17	0.25	0.28	0.19
	Example # 5	Example # 6	Example # 7	Example # 8
4.1 $\text{States}(C^{**})/\text{States}(Nb(C^*))$	1.00	0.84	0.53	0.81
4.2 $\text{Transitions}(C^{**})/\text{Transitions}(Nb(C^*))$	0.04	0.03	0.02	0.03
4.3 $\text{Max}(C^{**})/\text{Max}(Nb(C^*))$	$0.9 \cdot 10^{-4}$	$0.8 \cdot 10^{-4}$	$0.9 \cdot 10^{-4}$	$0.7 \cdot 10^{-4}$
4.4 $\text{Time}(C^{**})/\text{Time}(Nb(C^*))$	0.15	0.12	0.09	0.10

TABLE 4. Comparison between the computation of $Nb(C^*)$ and of C^{**} .

this direction can be found in the tool Pessoa [Pes09] which employs binary decision diagrams [Pac] as data structures to encode symbolic systems.

Acknowledgement. The first author would like to thank Paulo Tabuada for having inspired the idea of integration of control algorithms with the construction of the symbolic systems of the plant and the specification.

APPENDIX: NOTATION

The identity map on a set A is denoted by 1_A . Given two sets A and B , if A is a subset of B we denote by $1_A : A \hookrightarrow B$ or simply by ι the natural inclusion map taking any $a \in A$ to $\iota(a) = a \in B$. Given a function $f : A \rightarrow B$ the symbol $f(A)$ denotes the image of A through f , i.e. $f(A) := \{b \in B : \exists a \in A \text{ s.t. } b = f(a)\}$; if $C \subset A$ we denote by $f|_C$ the restriction of f to C , i.e. $f|_C(x) := f(x)$ for any $x \in C$. Given a relation $R \subseteq A \times B$, R^{-1} denotes the inverse relation of R , i.e. $R^{-1} := \{(b, a) \in B \times A : (a, b) \in A \times B\}$. A relation $R \subseteq A \times B$ is a preorder if it is reflexive, transitive but not symmetric. The symbols \mathbb{N} , \mathbb{Z} , \mathbb{R} , \mathbb{R}^+ and \mathbb{R}_0^+ denote the set of natural, integer, real, positive real, and nonnegative real numbers, respectively. Given a vector $x \in \mathbb{R}^n$, we denote by x_i the i -th element of x and by $\|x\|$ the infinity norm of x , we recall that $\|x\| = \max\{|x_1|, |x_2|, \dots, |x_n|\}$, where $|x_i|$ denotes the absolute value of x_i . Given a measurable function $f : \mathbb{R}_0^+ \rightarrow \mathbb{R}^n$, the (essential) supremum of f is denoted by $\|f\|_\infty$; we recall that $\|f\|_\infty = (\text{ess}) \sup \{\|f(t)\|, t \geq 0\}$; f is essentially bounded if $\|f\|_\infty < \infty$. Given $x \in \mathbb{R}^n$ and $\varepsilon \in \mathbb{R}^+$, the symbol $\mathcal{B}_\varepsilon(x)$ denotes the set $\{x \in \mathbb{R}^n : \|x\| \leq \varepsilon\}$ and the symbol $\mathcal{B}_{[\varepsilon]}(x)$ denotes the set $[-\varepsilon + x_1, x_1 + \varepsilon] \times [-\varepsilon + x_2, x_2 + \varepsilon] \times \dots \times [-\varepsilon + x_n, x_n + \varepsilon]$. It is readily seen that if $x \in \mathcal{B}_\varepsilon(y)$ and $y \in \mathcal{B}_{[\theta]}(z)$ then $x \in \mathcal{B}_{[\varepsilon+\theta]}(z)$. For any $A \subseteq \mathbb{R}^n$ and $\mu \in \mathbb{R}^+$, define $[A]_\mu = \{a \in A \mid a_i = k_i \mu, k_i \in \mathbb{Z}, i = 1, 2, \dots, n\}$. The set $[A]_\mu$ will be used as an approximation of the set A with precision $\mu/2$. For a given time $\tau \in \mathbb{R}^+$, define f_τ so that $f_\tau(t) = f(t)$, for any $t \in [0, \tau]$, and $f(t) = 0$ elsewhere; f is said to be locally essentially bounded if for any $\tau \in \mathbb{R}^+$, f_τ is essentially bounded. A continuous function $\gamma : \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$, is said to belong to class \mathcal{K} if it is strictly increasing and $\gamma(0) = 0$; γ is said to belong to class \mathcal{K}_∞ if $\gamma \in \mathcal{K}$ and $\gamma(r) \rightarrow \infty$ as $r \rightarrow \infty$. A continuous function $\beta : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+$ is said to belong to class \mathcal{KL} if, for each fixed s , the map $\beta(r, s)$ belongs to class \mathcal{K}_∞ with respect to r and, for each fixed r , the map $\beta(r, s)$ is decreasing with respect to s and $\beta(r, s) \rightarrow 0$ as $s \rightarrow \infty$.

REFERENCES

- [AD94] R. Alur and D.L. Dill. A theory of timed automata. *Theoretical Computer Science*, 126(2):183–235, 1994.
- [Ang02] D. Angeli. A Lyapunov approach to incremental stability properties. *IEEE Transactions on Automatic Control*, 47(3):410–421, 2002.
- [AS99] D. Angeli and E. Sontag. Forward completeness, unboundedness observability, and their lyapunov characterizations. *Systems and Control Letters*, 38:209–217, 1999.
- [BH06] C. Belta and L.C.G.J.M. Habets. Controlling a class of nonlinear systems on rectangles. *IEEE Transactions of Automatic Control*, 51(11):1749–1759, 2006.

- [BMP02] A. Bicchi, A. Marigo, and B. Piccoli. On the reachability of quantized control systems. *IEEE Transaction on Automatic Control*, April 2002.
- [CGP99] E.M. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 1999.
- [CL99] C. Cassandras and S. Lafortune. *Introduction to discrete event systems*. Kluwer Academic Publishers, Boston, MA, 1999.
- [CVWY92] C. Courcoubetis, M. Vardi, P. Wolper, and M. Yannakakis. Memory-efficient algorithms for the verification of temporal properties. *Formal Methods in System Design*, 1(2-3):275–288, 1992.
- [CW98] P.E. Caines and Y.J. Wei. Hierarchical hybrid control systems: A lattice-theoretic formulation. *Special Issue on Hybrid Systems, IEEE Transaction on Automatic Control*, 43(4):501–508, April 1998.
- [EFP06] M. Egerstedt, E. Frazzoli, and G. J. Pappas. *IEEE Transactions of Automatic Control*, 51(6), June 2006. Special Issue on Symbolic Methods for Complex Control Systems.
- [FJL02] D. Forstner, M. Jung, and J. Lunze. A discrete-event model of asynchronous quantised systems. *Automatica*, 38:1277–1286, 2002.
- [FPW98] G.F. Franklin, J.D. Powell, and M. Workman. *Digital Control of Dynamic Systems*. Ellis-Kagle Press, 1998.
- [Gir07] A. Girard. Approximately bisimilar finite abstractions of stable linear systems. In A. Bemporad, A. Bicchi, and G. Buttazzo, editors, *Hybrid Systems: Computation and Control*, volume 4416 of *Lecture Notes in Computer Science*, pages 231–244. Springer Verlag, Berlin, 2007.
- [GP07] A. Girard and G.J. Pappas. Approximation metrics for discrete and continuous systems. *IEEE Transactions on Automatic Control*, 52(5):782–798, 2007.
- [GPT10] A. Girard, G. Pola, and P. Tabuada. Approximately bisimilar symbolic models for incrementally stable switched systems. *IEEE Transactions of Automatic Control*, 55(1):116–126, January 2010.
- [HCS06] L.C.G.J.M. Habets, P.J. Collins, and J.H. Van Schuppen. Reachability and control synthesis for piecewise-affine hybrid systems on simplices. *IEEE Transaction on Automatic Control*, 51(6):938–948, 2006.
- [HKPV98] T.A. Henzinger, P.W. Kopke, A. Puri, and P. Varaiya. What’s decidable about hybrid automata? *Journal of Computer and System Sciences*, 57:94–124, 1998.
- [LPS00] G. Lafferriere, G.J. Pappas, and S. Sastry. O-minimal hybrid systems. *Mathematics of Control, Signals and Systems*, 13(1):1–21, March 2000.
- [Mil89] R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.
- [MRO02] T. Moor, J. Raisch, and S.D. O’Young. Discrete supervisory control of hybrid systems based on l-complete approximations. *Journal of Discrete Event Dynamic Systems*, 12:83–107, 2002.
- [Pac] CUDD: CU Decision Diagram Package. Electronically available at: <http://vlsi.colorado.edu/fabio/cudd/>.
- [Par81] D.M.R. Park. Concurrency and automata on infinite sequences. volume 104 of *Lecture Notes in Computer Science*, pages 167–183, 1981.
- [Pes09] Pessoa. Electronically available at: <http://www.cyphylab.ee.ucla.edu/pessoa>. 2009.
- [PGT08] G. Pola, A. Girard, and P. Tabuada. Approximately bisimilar symbolic models for nonlinear control systems. *Automatica*, 44:2508–2516, October 2008.
- [PPDT10] G. Pola, P. Pepe, M.D. Di Benedetto, and P. Tabuada. Symbolic models for nonlinear time-delay systems using approximate bisimulations. *Systems and Control Letters*, 59:365–373, 2010.
- [PT09] G. Pola and P. Tabuada. Symbolic models for nonlinear control systems: Alternating approximate bisimulations. *SIAM Journal on Control and Optimization*, 48(2):719–733, 2009.
- [Son98] E.D. Sontag. *Mathematical Control Theory*, volume 6 of *Texts in Applied Mathematics*. Springer-Verlag, New-York, 2nd edition, 1998.
- [TA99] S. Tripakis and K. Altisen. On-the-fly controller synthesis for discrete and dense-time systems. In *World Congress on Formal Methods in the Development of Computing Systems*, volume 1708 of *Lecture Notes in Computer Science*, pages 233 – 252. Springer Verlag, Berlin, September 1999.
- [Tab08] P. Tabuada. An approximate simulation approach to symbolic control. *IEEE Transactions on Automatic Control*, 53(6):1406–1418, 2008.
- [Tab09] P. Tabuada. *Verification and Control of Hybrid Systems: A Symbolic Approach*. Springer, 2009.
- [Tii09] Yuichi Tazaki and Jun ichi Imura. Discrete-state abstractions of nonlinear systems using multi-resolution quantizer. In R. Majumdar and P. Tabuada, editors, *Hybrid Systems: Computation and Control*, volume 5469 of *Lecture Notes in Computer Science*, pages 351–365. Springer Verlag, Berlin, 2009.
- [TP06] P. Tabuada and G.J. Pappas. Linear Time Logic control of discrete-time linear systems. *IEEE Transactions on Automatic Control*, 51(12):1862–1877, 2006.
- [YB09] B. Yordanov and C. Belta. Temporal logic control of discrete-time piecewise affine systems. In *48th IEEE Conference on Decision and Control*, pages 3182–3187, Shanghai, P.R. China, December 2009.
- [ZPT10] M. Zamani, G. Pola, and P. Tabuada. Symbolic models for unstable nonlinear control systems. In *Proceedings of the 2010 American Control Conference*, Baltimore, Maryland, USA, June 2010. To appear.

¹ DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING, CENTER OF EXCELLENCE DEWS, UNIVERSITY OF L'AQUILA, POGGIO DI ROIO, 67040 L'AQUILA, ITALY

E-mail address: {giordano.pola,alessandro.borri,mariadomenica.dibenedetto}@univaq.it